IndicSTR12: A Dataset for Indic Scene Text Recognition

Harsh Lunia(✉), Ajoy Mondal, and C. V. Jawahar

Centre for Vision Information Technology, International Institute of Information Technology,
Hyderabad 500032, India
harsh.lunia@research.iiit.ac.in,
{ajoy.mondal,jawahar}@iiit.ac.in
http://cvit.iiit.ac.in/research/projects/cvit-projects/indicstr

Abstract. The importance of Scene Text Recognition (STR) in today's increasingly digital world cannot be overstated. Given the significance of STR, data-intensive deep learning approaches that auto-learn feature mappings have primarily driven the development of STR solutions. Several benchmark datasets and substantial work on deep learning models are available for Latin languages to meet this need. On more complex, syntactically and semantically, Indian languages spoken and read by 1.3 billion people, there is less work and datasets available. This paper aims to address the Indian space's lack of a comprehensive dataset by proposing the largest and most comprehensive real dataset - IndicSTR12 - and benchmarking STR performance on 12 major Indian languages (Assamese, Bengali, Odia, Marathi, Hindi, Kannada, Urdu, Telugu, Malayalam, Tamil, Gujarati, and Punjabi). A few works have addressed the same issue, but to the best of our knowledge, they focused on a small number of Indian languages. The size and complexity of the proposed dataset are comparable to those of existing Latin contemporaries, while its multilingualism will catalyse the development of robust text detection and recognition models. It was created specifically for a group of related languages with different scripts. The dataset contains over 27000 word-images gathered from various natural scenes, with over 1000 word-images for each language. Unlike previous datasets, the images cover a broader range of realistic conditions, including blur, illumination changes, occlusion, non-iconic texts, low resolution, perspective text etc. Along with the new dataset, we provide a high-performing baseline on three models: PARSeq (Latin SOTA), CRNN, and STARNet.

Keywords: Scene Text Recognition · Indian Languages · Synthetic Dataset · Photo-OCR · OCR · Multi-lingual · Indic Scripts · Real Dataset

1 Introduction

Language has enabled people world around to exchange and communicate. Different communities have recognized the importance of the same, which becomes evident by the diversity of languages across the globe. The textual representation of language, on the other hand, significantly broadens the scope of transfer. Semantically rich writing

M. Coustaty and A. Fornés (Eds.): ICDAR 2023 Workshops, LNCS 14193, pp. 233–250, 2023.
https://doi.org/10.1007/978-3-031-41498-5_17

found in the wild has powerful information that can substantially aid in understanding the surrounding environment in the modern era. Textual information found in the wild is used for various tasks, including image search, translation, transliteration, assistive technologies (particularly for the visually impaired), autonomous navigation, and so on. The issue of automatically reading text from photographs or frames of a natural environment is referred to as Scene Text Recognition (STR) or Photo-OCR. This problem is typically subdivided into two sub-problems: Scene text detection, which deals with locating text within a picture, and cropped word image recognition. Our work addresses the second sub-problem: recognizing the text in a clipped word image.

Traditionally, OCR has focused on reading printed or handwritten text in documents. However, as capturing devices such as mobile phones and video cameras have proliferated, scene text recognition has become critical and a problem whose solution holds promise for furthering the resolution of other downstream tasks. Although there has been considerable progress in STR, it has some unique issues, i) varying backgrounds in natural scenes, ii) varying script, font, layout, and style, and iii) text-related image flaws such as blurriness, occlusion, uneven illumination, etc. Researchers have attempted to address the aforementioned issue by amassing datasets specifically tailored to a given problem, each of which has some distinguishing features and represents a subset of challenges encountered in real-world situations. For example, [28] has perspective text, [40] has blurred text, and [29] has curved text.

Rather than designing and testing manually created features, nearly all current solutions rely on deep learning techniques to automate feature learning. Because of the data-intensive nature of these models, it has become standard practice to train the models on synthetically generated data that closely resemble real-world circumstances and test the trained models on difficult-to-obtain real datasets. STR solutions for Latin languages such as English have made significant progress. However, Latin STR having reached a certain level of maturity, has begun to train solely on available real datasets [3] to achieve nearly comparable performance compared to a mix of synthetic and real. Using an already available diverse set of public real datasets totaling almost 0.3 million image instances for English was a significant factor that led to similar results.

Because Indian language scripts are visually more complex, and their output space is much larger than English languages, not all Latin STR models can mimic performance in Indian STR solutions [24]. STR solutions have not progressed in the case of Indian languages, which are spoken by 18% of the world population, due to a lack of real datasets and models that are better equipped to handle the inherent complexities of the languages. Non-Latin languages have made less progress, and existing Latin STR models need to generalize better to different languages [8].

Contribution. Given the need for more data for non-Latin languages, particularly Indian languages, our work attempts to address the issue by contributing as follows:

1. We propose a real dataset (Fig. 1 (left)) for 12 Major Indian Languages, namely - Assamese, Bengali, Odia, Marathi, Hindi, Kannada, Urdu, Telugu, Malayalam, Tamil, Gujarati and Punjabi - wherein Malayalam, Telugu, Hindi, and Tamil word-images have been taken from [23] and [12]. Since the number of word instances proposed by [12] for Gujarati was less than 1000 work image instances, we augment

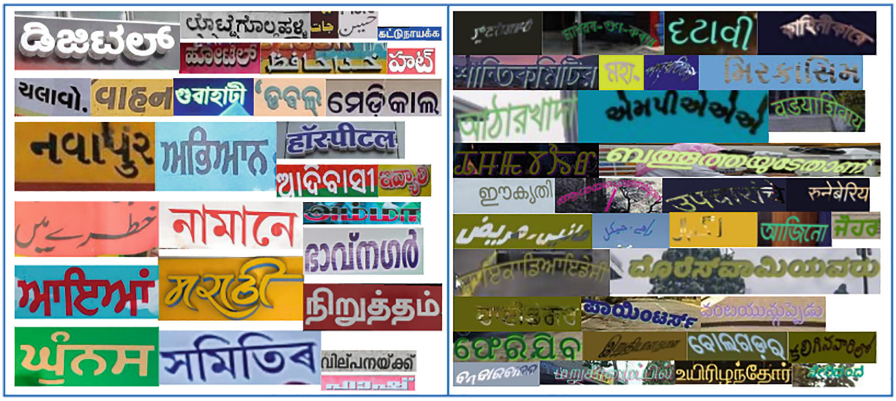


Fig. 1. Samples from IndicSTR12 Dataset: Real word-images (left); Synthetic word-images (right)

the proposed Gujarati instances to achieve numbers comparable to other languages in the proposed dataset.

2. We propose a synthetic dataset for all 13 Indian languages (Fig. 1 (right)), which will help the STR community progress on multi-lingual STR, in effect similar to SynthText [13] and MJSynth [16].
3. Finally, we compare the performance of three STR models - PARSeq [5], CRNN, and STARNet [20] - on all 12 Indian languages, some of which have no previous benchmark to compare with. The effectiveness of these models on the IndicSTR12 dataset and other publicly accessible datasets supports our dataset's claim that it is challenging (Table 6).
4. By simultaneously training on multiple languages' real datasets, we demonstrate how multi-lingual recognition models can aid models in learning better, even with sparse real data.

2 Related Works

Scene text recognition (STR) models use CNNs to encode image features. For decoding text out of the learnt image features in a segmentation freeway, it relies either on Connectionist Temporal Classification (CTC) [11] or encoder-decoder framework [37] combined with attention mechanism [4]. The CTC-based approaches [20,34] treat images as a sequence of vertical frames and combine prediction per frame based on a rule to generate the whole text. In contrast, the encoder-decoder framework [21] uses attention to align input and output sequences. There has been work on both CTC and attention-based models for STR. DTRN [15] is the first to use CRNN models, a combination of CNNs with RNNs stacked on them, to generate convolutional feature slices to be fed to RNNs. Using attention, [21] performs STR based on encoder-decoder model wherein the encoder is trained in binary constraints to reduce computation cost. Work on datasets of varying complexity has been done to promote research on STR for challenging scenarios as well. A few challenging ones in terms of occlusion, blur, small and multi orientation word-images are ICDAR 2015 [17], Total-Text [9], LSVT [35,36].

Indian Scene Text Recognition. The lack of annotated data is a hurdle, especially in the case of Indian languages, in realizing the success achieved in the case of Latin STR solutions. There has been attempt to address the data scarcity problem over the years in a scattered and very language-specific way for Indian languages. [7] is the first work to propose an Urdu dataset and benchmark STR performance on Urdu text. It contains 2,500 images, giving 14,100 word-images. The MLT-17 dataset [27] contains 18k scene images in multiple languages, including Bengali. Building on top of it, MLT-19 [26] contains 20k scene images in multiple languages, including Bengali and Hindi. To our knowledge, this is currently the only multilingual dataset, and it supports ten different languages. [23] trains a CRNN model on synthetic data for three Indian languages: Malayalam, Devanagari (Hindi), and Telugu. It also releases an IIIT-ILST dataset for mentioned three languages for testing, reporting a WRR of 42.9%, 57.2% and 73.4% in Hindi, Telugu, and Malayalam, respectively. [6] proposes a CNN and CTC-based method for script identification, text localization, and recognition. The model is trained and tested on MLT 17 dataset, achieving 34.20% WRR for Bengali. An OCR-on-the-go model [30] obtained a WRR of 51.01% on the IIIT-ILST Hindi dataset and a CRR of 35% on a multi-lingual dataset containing 1000 videos in English, Hindi, and Marathi. [12] explored transfer learning among Indian languages as an approach to increase WRR and proposed a dataset of natural scene images in Gujarati and Tamil to test the hypothesis further. It achieved a WRR gain of 6, 5 and 2% on the IIIT-ILST dataset and a WRR of 69.60% and 72.95% in Gujarati and Tamil respectively.

3 Datasets and Motivation

3.1 Synthetic Dataset

Table 1. Statistics of Synthetic Data

Language	Vocabulary Size	Mean, Std	Min, Max	Fonts
Gujarati	106,551	5.96 , 1.85	1 , 20	12
Urdu	234,331	6.39 , 2.20	1 , 42	255
Punjabi	181,254	6.45 , 2.18	1 , 31	141
Manipuri (Meitei)	66,222	6.92 , 2.49	1 , 29	24
Assamese	77,352	7.08 , 2.70	1 , 46	85
Odia	149,681	8.00 , 3.00	1 , 37	30
Bengali	449,986	8.53 , 3.00	1 , 38	85
Marathi	180,278	8.64 , 3.43	1 , 44	218
Hindi	319,982	8.76 , 3.20	1 , 50	218
Telugu	499,969	9.75 , 3.38	1 , 50	62
Tamil	399,999	10.75 , 3.64	1 , 35	158
Kannada	499,972	10.72 , 3.87	1 , 41	30
Malayalam	320,000	14.30 , 5.36	1 , 53	101

It has been an accepted practice in the community to train an STR model on a large synthetically generated dataset since [16] trained a model on 8 Million synthetically generated English word-images called MJSynth. This trend, as [2] points, is due to the high cost of annotating real data. Another synthetic dataset widely in use for English language is SynthText [13]. On the Indian language side, there have been some works like [12, 23] which use synthetic datasets for a total of 6 Indian Languages. However, like real dataset scenario, a comprehensive synthetic dataset for all 12 major Indian languages is absent. We extend the previously referred work and propose a synthetic dataset for all 12 Major Indian languages by following the same procedure as [23]. The word images are rendered by randomly sampling words from a vocabulary of more than 100K words for each language (except for Assamese) and rendering them using freely available Unicode fonts Table 1. Each word image is first rendered in the foreground layer by varying font, size, and stroke thickness, color, kerning, rotation along the horizontal line, and skew. This is followed by the applying a random perspective projective transformation to the foreground layer and, consequently, a blending of the same with a random crop from a natural scene image taken from Places365 dataset [42]. Lastly, the foreground image is alpha composed with a background image which can either be a random crop from a natural scene image or one having a uniform color. The synthetic dataset proposed has more than 3 Million word images per language. For benchmarking STR performance, we have followed the same procedure as [12], using 2 Million word images for training the network and 0.5 Million for validation and testing.

3.2 Real Dataset

Table 2. Usage Statistics and General Information of Official Indian Languages not part of IndicSTR12 Dataset

Language	Script	Usage	Family
Bodo	Devanagari	1.4M	Sino-Tibetan
Kashmiri	Arabic & Devanagari	11M	Indo-Aryan
Dogri	Devanagari	2.6M	Indo-Aryan
Konkani	Devanagari	2.3M	Indo-Aryan
Maithali	Devanagari	34M	Indo-Aryan
Sindhi	Arabic & Devanagari	32M	Indo-Aryan
Santhali	Ol Chiki	7.6M	Austroasiatic
Nepali	Devanagari	25M	Indo-Aryan

According to the Census 2011 report on Indian languages [10], India has 22 major or scheduled languages with a significant volume of writing. All major Indian languages can be classified into four language families: Indo-Aryan, Dravidian, Sino-Tibetan, and Austro-Asian (listed in decreasing order of usage). Sanskrit, Bodo, Dogri, Kashmiri, Konkani, Maithili, Nepali, Santali, and Sindhi are among the languages not covered by the IndicSTR12 dataset Table 2. As a classical language, Sanskrit has a long history of heavily influencing all of the subcontinent's languages. It is now widely taught at the secondary level, but its use is limited to ceremonial and ritualistic purposes with no first-language speakers. Other languages that have been left out either have scripts that are similar to one of the included languages[1] or have minimal usage in the domain of scene text in natural settings[2]. According to the 2011 census report [10], the included languages cover 98% of the subcontinent's spoken language. IndicSTR12 is an extension of IIIT-ILST [23] and [12], which cover Telugu, Malayalam, Hindi, Gujarati, and Tamil, respectively. There has been no addition of images for any of the mentioned languages, except for Gujarati, which had less than 1000 word-images.

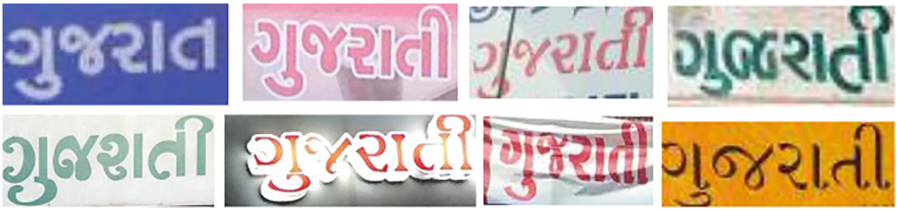


Fig. 2. IndicSTR12 Dataset: Font Variations for the same word - Gujarati or Gujarat

IndicSTR12 Curation Details. All the images have been crawled from Google Images using various keyword-based searches to cover all the daily avenues wherein Indic language text can be observed in natural settings. To mention some - Wall paintings, railway stations, Signboards, shop/temple/mosque/gurudwara name-boards, advertisement banners, political protests, house plates, etc. Because they have been crawled from a search engine, they come from various sources, offering a wide range of conditions under which images were captured. Curated images have blur, non-iconic/iconic text, low-resolution, occlusion, curved text, perspective projections due to non-frontal viewpoints, etc Figs. 3 and 2.

IndicSTR12 Annotation Details. All the words within the image are annotated with four-corner point annotation as done in [26] to capture both horizontal and curved word structures of scene texts. The annotators were encouraged to follow the reading direction and include as little background space as possible. To further ensure the quality of annotation, all the annotated data was reviewed by another entity to ensure proper removal/correction of empty or wrong labels. The reviewers were also tasked with further classifying each word instance into the three categories of Oriented Text, Low-resolution/Smaller Text, and Occluded Text. This will enable community members to assess which areas of a model's performance require special consideration. There are at least 1000 word images per language and their corresponding labels in Unicode. The dataset can also be used for the problem of script identification and scene text detection.

[1] Bodo, Dogri, Kashmiri, Konkani, Maithili, Nepali, and Sindhi.

[2] Santali.

3.3 Comparision with Existing Datasets

Table 3. Statistics of Various Public STR Real Dataset

Dataset	Word Images (train/test)	Language	Features	Tasks[a]
IIIT5K-Words [25]	2K/3K	English	Regular	R
SVT [40]	211/514	English	Regular, blur, low resolution	D,R
ICDAR2003 [22]	1157/1111	English	Regular	D,R
ICDAR2013 [18]	3564/1439	English	Regular, Stroke Labels	D,R
ICDAR2015 [17]	4468/2077	English	Irregular, Blur, Small	D,R
SVT Perspective [28]	0/639	English	Irregular, Perspective Text	R
MLT-19 [26]	89K/102K	Multi-Lingual[b]	Irregular	D,R
MTWI [14]	141K/148K	Chinese, English	Irregular	R
LSVT [35,36]	30K/20K	Chinese, English	Irregular, multi-oriented	D, R
MLT-17 [27]	85K/11K	Multi-Lingual[c]	Irregular	D,R
Urdu-Text [7]	14100	Urdu	Irregular, Noisy	D,R
CUTE80 [29]	0/288	English	Irregular, Perspective Text, Low Resolution	D,R
IndicSTR12 (Ours)[d]	20K/7K	Multi-Lingual[e]	Irregular, Low Resolution, Blur, Occlusion, Perspective Text	D,R

[a] 'D' Stands for Detection and 'R' for Recognition

[b] Arabic, Bangla, Chinese, Devanagari, English, French, German, Italian, Japanese, and Korean

[c] Arabic, Bangla, Chinese, English, French, German, Italian, Japanese, and Korean

[d] Extends IIIT-ILST [12,23]

[e] Assamese, Bengali, Odia, Marathi, Hindi, Kannada, Urdu, Telugu, Malayalam, Tamil, Gujarati and Punjabi

Real datasets were initially utilised for fine-tuning models trained on synthetic datasets and evaluating trained STR models because the majority of real data sets only contain thousands of word-images. There are many datasets available for English Languages that address a variety of difficulties, but the community has only now begun to consider training models with real data [3]. Broadly a real dataset can be seen as regular and irregular types. Regular datasets have most word-images that are iconic (frontal), horizontal and a small portion of distorted samples. In contrast, the majority of the word-images in irregular datasets are perspective text, low-resolution, and multi-oriented. The high variation in text instances makes these difficult for STR. For examples and more information, please refer to Table 3. Our Dataset has been curated with the goal of catering to both regular and irregular samples. Being extracted from the Google search engine via various keywords generally associated with scene and text, it tries to cover a wide range of natural scenarios, mostly seen in Indian Scene texts. The classification of word-images into categories was done to allow Indian STR solutions to assess their current standing on regular texts and, as the solutions advance, to provide a challenging subset to further refine the prediction models.

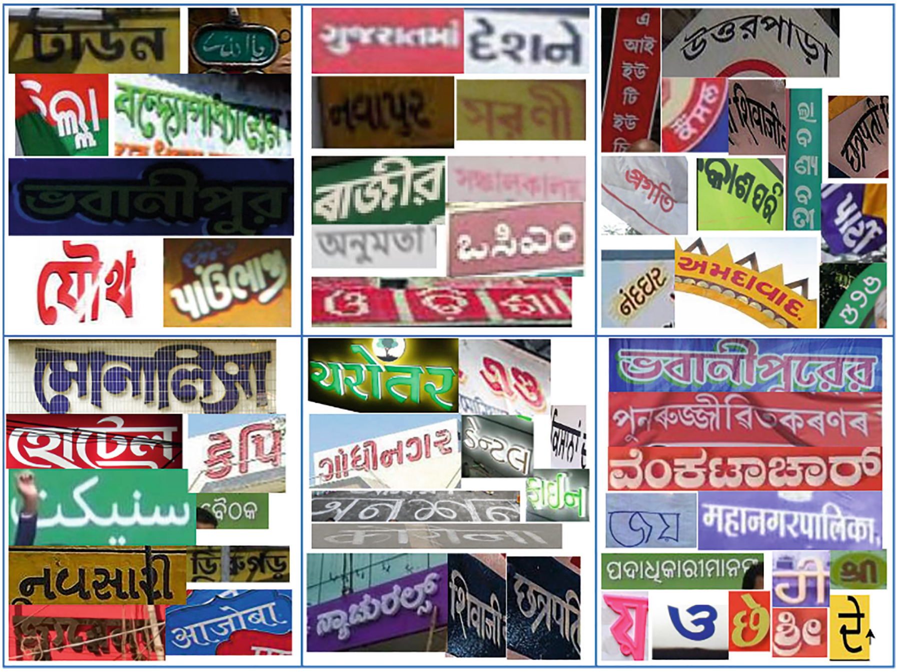


Fig. 3. IndicSTR12 Dataset Variations, clockwise from Top-Left: Illumination variation, Low Resolution, Multi-Oriented - Irregular Text, Variation in Text Length, Perspective Text, and Occluded.

4 Models

This section explains the models used to benchmark STR performance on 12 Indian languages in IndicSTR12 dataset. Three models were picked up to benchmark the STR performance - PARSeq [5] is the current state-of-the-art Latin STR model, CRNN [31] which has low accuracy than a lot of current models but is widely chosen by the community for practical usage because it's lightweight and fast. Another model called STAR-Net [20], extracts more robust features from word-images and performs an initial distortion correction, is also used for benchmarking. This model has been taken up to maintain consistency with the previous works on Indic STR [12,23].

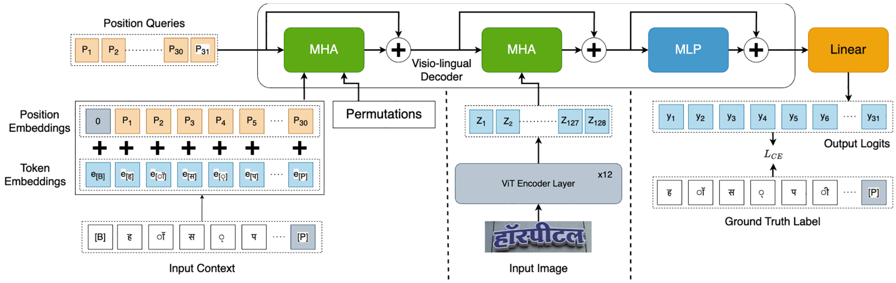


Fig. 4. PARSeq architecture. [B] and [P] begin the sequence and padding tokens. T = 30 or 30 distinct position tokens. L_{CE} corresponds to cross entropy loss.

PARSeq. Figure 4 PARSeq is a transformer-based model which is trained using Permutation Language Modeling (PLM). Multi-head Attention [39] is extensively used, MHA(q, k, v, m), where q, k, v and m refer to query, key, value and optional attention mask. The model follows an encoder-decoder architecture wherein the encoder stack has 12 encoder blocks while the decoder has only a single block.

The model uses 12 vision transformer encoder blocks each containing 1 self-attention MHA module. The image $x \in \mathbb{R}^{\mathbb{W}\times\mathbb{H}\times\mathbb{C}}$ is tokenized evenly into $p_w \times p_h$ patches which are in-turn projected into d_{model} - dimensional tokens using an embedding matrix $\boldsymbol{W}^p \in \mathbb{R}^{p_w p_h C \times d_{model}}$. Position embeddings are then added to tokens before sending them to the first ViT encoder block. All output tokens $\boldsymbol{z}$ are used as input to the decoder:

$$\boldsymbol{z} = Enc(x) \in \mathbb{R}^{\frac{WH}{p_w p_h} \times d_{model}}$$

The Visio-lingual Decoder used is a pre-layerNorm transformer decoder with two MHAs. The first MHA requires position tokens, $\boldsymbol{p} \in \mathbb{R}^{(T+1)\times d_{model}}$ (T being context length), context embeddings, $\boldsymbol{c} \in \mathbb{R}^{(T+1)\times d_{model}}$, and attention mask, $\boldsymbol{m} \in \mathbb{R}^{(T+1)\times(T+1}$. The position token captures the target position to be predicted and decouples the context from the target position, enabling the model to learn from permutation language modeling. The attention masks vary at different use points. During training, they are based on permutations, while during inference it is a left-to-right look-ahead mask. Transformers process all tokens in parallel, therefore to enforce the condition of past tokens having no access to future ones, attention masks are used. PLM, which in theory requires the model to train on all $T!$ factorizations, in practice is achieved by using attention masks to enforce some subset K of $T!$ permutations.

$$\boldsymbol{h}_c = \boldsymbol{p} + MHA(\boldsymbol{p}, \boldsymbol{c}, \boldsymbol{c}, \boldsymbol{m}) \in \mathbb{R}^{(T+1} \times d_{model}$$

The second MHA is used for image-position attention where no attention mask is used.

$$\boldsymbol{h}_i = \boldsymbol{h}_c + MHA(\boldsymbol{h}_c, \boldsymbol{z}, \boldsymbol{z} \in \mathbb{R}^{(T+1} \times d_{model})$$

The last decoder hidden state is used to get the output logits $\boldsymbol{y} = Linear(\boldsymbol{h}_{dec} \in \mathbb{R}^{(T+1)\times(S+1)}$ where S is the size of the character set and an addition of 1 is due to end of sequence token $[\boldsymbol{E}]$.

The decoder block can be represented by:

$$\boldsymbol{y} = Dec(\boldsymbol{z}, \boldsymbol{p}, \boldsymbol{c}, \boldsymbol{m}) \in \mathbb{R}^{(T+1)\times(S+1)}$$

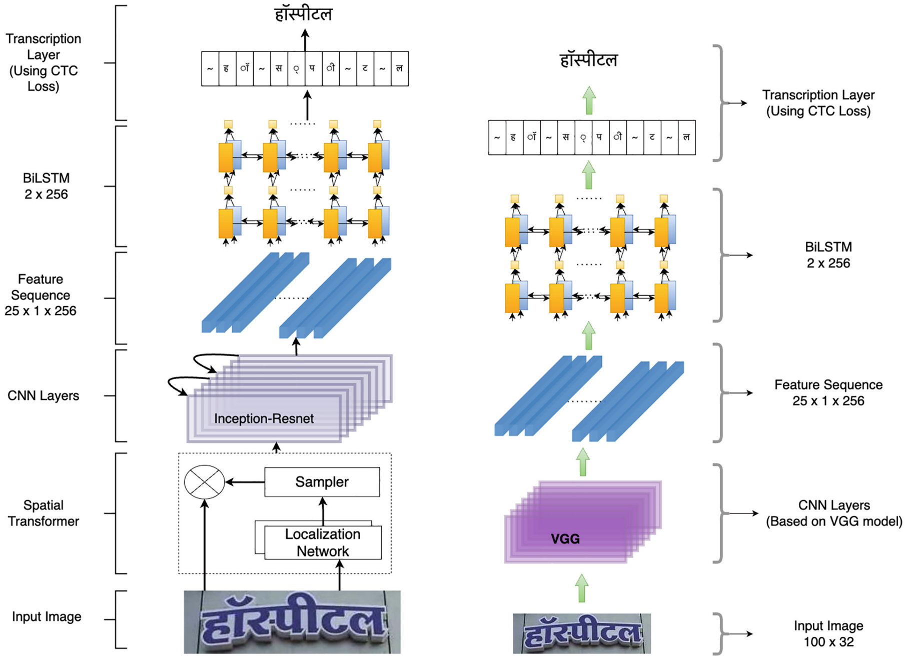


Fig. 5. STARNet model (left) and CRNN model (right)

CRNN. CRNN is a combination of CNN and RNN, as shown in Fig. 5 (right). The model primarily can be viewed as a combination of 3 components - (i) an encoder, here a standard VGG model [32], to extract features from word-image, (ii) a decoder consisting of RNN and lastly, (iii) a Connectionist Temporal Classification (CTC) layer which aligns decoded sequence to ground truth. The CNN encoder is made of 7 layers to extract the feature maps. For RNN, it uses a two-layer BiLSTM model, each with a hidden size of 256 units. During training, the CTC layer provides non-parameterized supervision to ensure that predictions match the ground truth. All of our experiments used a PyTorch implementation of the model by [31].

STARNet. STARNet in Fig. 5 (left), like CRNN, consists of three components: an encoder that is a CNN-based model, a RNN-based decoder, and a Connectionist Temporal Classification (CTC) layer to align the decoded sequence with the ground truths.

However, it differs from CRNN in two ways: it performs initial distortion correction using a spatial transformer, and its CNN is based on an inception ResNet architecture [38], which can extract more robust features required for STR.

5 Experiments

Each STR model is trained on 2 million and tested on 0.5 million synthetic word-images. To further adapt the model to real-world word-images, it is trained and tested on the proposed IndicSTR12. We use 75% of the word-images for training and 25% for testing in each language.

All PARSeq models are trained on dual-GPU platforms with Pytorch DDP for 20–33 epochs and 128 batch size. In conjunction with the 1cycle learning rate scheduler [33], the Adam optimizer [19] is used. As in the PARSeq model, we use K = 6 permutations with mirroring for PLM and an 8 × 4 patch size for ViTSTR. The maximum label length for the transformer-based PARSeq model is determined by the vocabulary used to create the synthetic dataset. We avoid using any data augmentation on synthetic datasets in accordance with community practise [3].

Using a spatial transformer, the STARNet model transforms a resized input image of 150 × 18 to 100 × 32. Both STARNet and CRNN encoders accept images of size 100 × 32, and the output feature maps are of size 23 × 1 × 256. All CRNN and STARNet models are trained on 2 Million synthetic images on a batch size of 32 and with ADADELTA [41] optimizer for stochastic gradient descent. The number of epochs is fixed at 15. For each language, the models are tested on 0.5 million synthetic images.

We also run experiments on other Indic datasets, including MLT-17 [27] for Bengali (referred to as Bangla in MLT-17), MLT-19 [26] for Hindi, and Urdu-Text [7] for Urdu. For all other public real datasets, we finetune on the train split and test our models on the test split (if no test splits are available, the proposer's val split is used).

6 Result and Analysis

6.1 Benchmarking

In this section, we first list and compare the respective models' performance on a synthetic dataset, then on our proposed real dataset.

Performance on Synthetic Dataset. Table 4 shows the CRR and WRR for each of the 13 languages achieved by various models. According to the data, the PARSeq model [5] has clearly outperformed other models in terms of WRR and CRR in all 13 languages. This gain can be attributed to it being an attention-based model and using Permutation Language Modeling to further capture the context. Attention models have been shown to outperform CTC-based models in general by Latin solutions; the same holds true for the Indic language in the case of synthetic data points.

Table 4. Performance on Synthetic Data

Language	CRNN		STARNet		PARSeq	
	CRR	WRR	CRR	WRR	CRR	WRR
Kannada	83.44	48.69	90.71	66.13	**97.26**	**87.92**
Odiya	89.94	66.57	95.05	81.00	**98.46**	**93.27**
Punjabi	88.88	66.11	93.86	78.34	**97.08**	**87.89**
Urdu	76.83	39.90	86.51	58.35	**95.26**	**80.48**
Marathi	86.91	59.87	93.55	76.32	**99.08**	**95.82**
Assamese	88.86	67.90	94.76	82.93	**99.21**	**96.85**
Manipuri (Meitei)	89.83	67.57	94.71	80.74	**98.79**	**94.76**
Malayalam	85.48	48.42	93.23	69.28	**98.71**	**92.27**
Telugu	81.12	43.75	89.54	62.23	**96.31**	**83.4**
Hindi[a]	89.83	73.15	95.78	83.93	**99.13**	**95.61**
Bengali[a]	91.54	70.76	95.52	82.79	**98.39**	**92.56**
Tamil[a]	82.86	48.19	95.40	79.90	**97.88**	**90.31**
Gujarati[a]	94.43	81.85	97.80	91.40	**98.82**	**95.25**

[a] Values for CRNN and STARNet have been taken from Guna *et al.* [12] as the training parameters and synthetic data generator were same.

Performance on Real Dataset

IndicSTR12 Dataset: The CRR and WRR numbers for the three models on the IndicSTR12 dataset are listed in the Table 5. Because the dataset is an extension of [12] and IIIT-ILST [23], the Hindi and Tamil numbers have been directly quoted from their work. We conducted separate experiments for Malayalam and Telugu (also covered by the two papers) because the cited works used a larger number of synthetic data to achieve higher accuracies. This was done to make a more accurate comparison with other languages' performance and to accurately gauge the models' performance on real data. Furthermore, the existing data for Gujarati was less than the required minimum of 1000 word-images per language, so this extension also supplements the Gujarati dataset.

A careful examination of the WRR and CRR numbers reveals that the PARSeq model outperforms in almost all cases where the real dataset is large enough, say greater than 1500. In a few cases, PARSeq falls short of the other two due to a lack of word-image instances for the model to train on.

Other Public Dataset: The models' overall performance, as shown in Table 6, followed a similar pattern to that of the IndicSTR12 dataset. However, in contrast to trend seen for IndicSTR12, the performance of the STARNet models is comparable to that of PARSeq and somewhat better for MLT-19 Hindi and Urdu Text. Importantly, all models can be seen to achieve WRR substantially higher than in the case of the IndicSTR12 dataset if the number of word-images is similar. This trend is most pronounced for Urdu Text (refer Fig. 6, where 10% of the original dataset roughly equals the number

Table 5. Performance on IndicSTR12

Language	CRNN		STARNet		PARSeq		Word-images
	CRR	WRR	CRR	WRR	CRR	WRR	
Kannada	78.79	52.43	82.59	59.72	**88.64**	**63.57**	1074
Odiya	80.39	54.74	86.97	66.30	**89.13**	**71.30**	3650
Punjabi	83.15	68.85	84.93	62.5	**92.68**	**78.70**	3887
Urdu	63.68	26.7	74.60	41.48	**76.97**	**44.19**	1375
Marathi	70.79	50.96	83.73	58.65	**86.74**	**63.50**	1650
Assamese	59.25	43.02	80.97	51.83	**81.36**	**52.70**	2154
Malayalam	77.94	53.12	84.97	**70.09**	**90.10**	68.81	807
Telugu	78.07	58.12	85.52	63.44	**92.18**	**71.94**	1211
Hindi[a]	**78.84**	46.56	78.72	**46.60**	76.01	45.14	1150
Bengali	59.86	48.21	80.26	57.70	**83.08**	**62.04**	3520
Tamil[a]	75.05	59.06	**89.69**	**71.54**	87.56	67.35	2536
Gujarati (New)[b]	52.22	23.05	**75.75**	41.80	74.49	**45.10**	1021
Gujarati	53.34	42.58	74.82	51.56	**85.02**	**60.61**	922

[a] Values for CRNN and STARNet have been taken from Guna *et al.* [12] as the training parameters and real data were same.
[b] Excluding [12] data

in IndicSTR12), MLT-19 Bengali, and MLT-17 Bengali. This further demonstrates that IndicSTR12 is more challenging due to all the irregular samples and the fact that most of the images in the MLT-17, MLT-19, and Urdu Text dataset are frontal captures or regular in form. In the case of the Urdu Text dataset, the models achieved nearly identical performance utilizing just half of the dataset.

Table 6. Performance on Other Public Datasets

Dataset	Word-images	CRNN		STARNet		PARSeq	
	Train/Test	CRR	WRR	CRR	WRR	CRR	WRR
MLT-17 (Bengali)	3237/713	79.98	55.30	85.24	65.73	**88.72**	**71.25**
MLT-19 (Bengali)	3935/-	82.80	59.51	89.46	71.25	**90.10**	**72.59**
MLT-19 (Hindi)	3931/-	86.48	67.90	91.00	**75.97**	**91.80**	75.91
Urdu -Text	12076/1480	93.33	82.45	97.91	**94.26**	**97.93**	93.92

Error Analysis. Some failure cases were found after analyzing the PARSeq model's ViT encoder attention maps for prediction mistakes (Fig. (7)) using the method proposed by [1]. The PARSeq model can recognize the textual region even in irregular word-image examples, but its predictions for the text are severely inaccurate for low-resolution images (Fig. (7a)) and only somewhat reliable for rotated or curved text

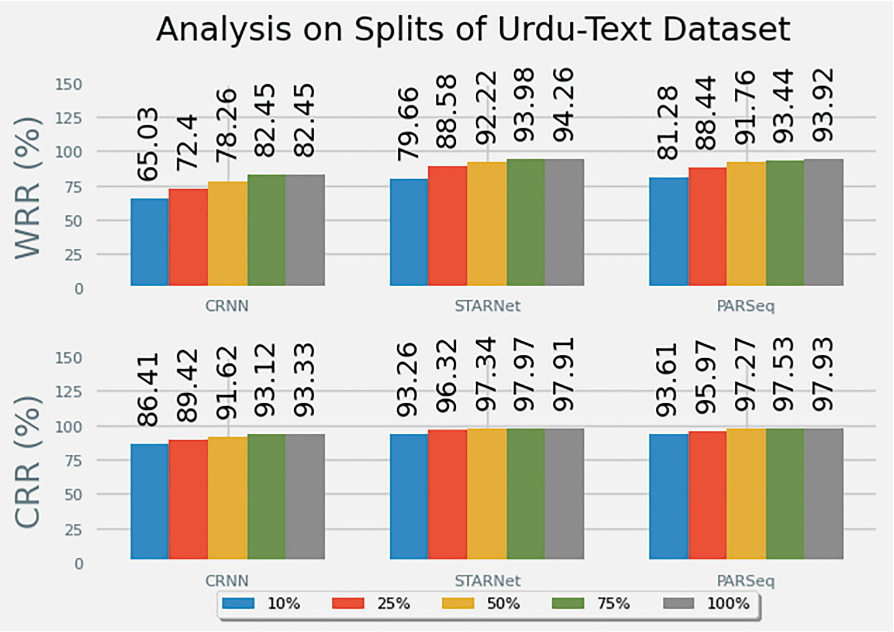


Fig. 6. Urdu Splits Analysis: Effect of training samples on STR models performance

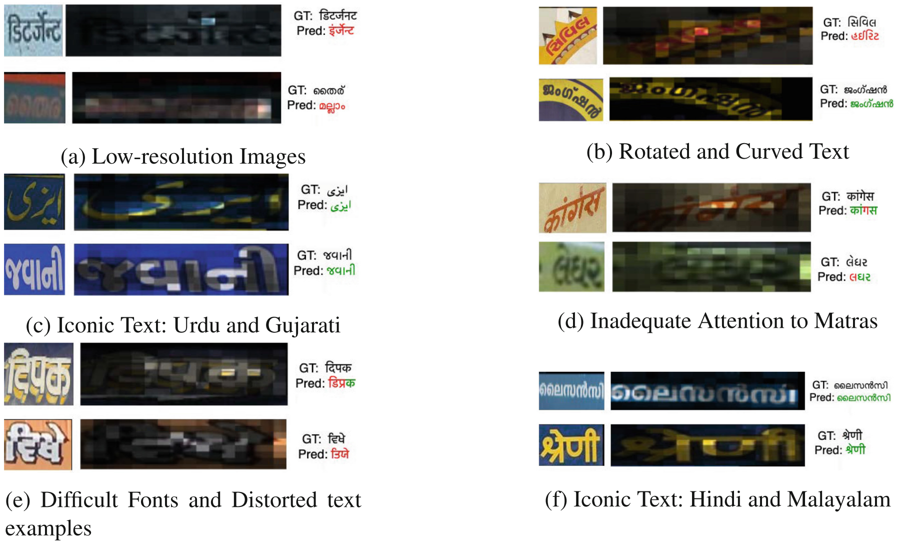
(Fig. (7b)). Additionally, because lower and upper matra are not given adequate consideration, the model misses a lot of accurate predictions (Fig. (7d)). Another notable instance of failure is when distorted text or unusual fonts cause shadows or border thickness to be seen as a different character (Fig. (7e)). The model does reasonably well when dealing with typical iconic texts (Fig. (7f)) and does well overall when dealing with lengthy texts.

6.2 Multi-lingual Training

Table 7. Multi-Lingual Training

Lang.(s) Trained On	Lang. Tested On	PARSeq	
		CRR	WRR
Hindi	Hindi	76.01	45.14
Hindi-Gujarati	Hindi	**76.41**	**49.65**
Gujarati	Gujarati	74.49	45.51
Hindi-Gujarati	Gujarati	**75.68**	**45.70**

The community has investigated transfer learning for STR models [12] as well as multilingual models in the case of OCR [24] since there few real training examples available.



(a) Low-resolution Images

(b) Rotated and Curved Text

(c) Iconic Text: Urdu and Gujarati

(d) Inadequate Attention to Matras

(e) Difficult Fonts and Distorted text examples

(f) Iconic Text: Hindi and Malayalam

Fig. 7. Error Analysis using Attention Maps

It is warranted as Indic language groups share a syntactic and semantic commonality. We looked into the multi-lingual model perspective here for Indic Languages in STR to provide a demonstrative example. In order to have a fair comparison, since single language models are trained on 2M synthetic datasets and finetuned on their respective real images, we trained our multi-lingual model on the same number of synthetic images-1M Hindi and 1M Gujarati-and then finetuned on a combined Hindi-Gujarati real images dataset, to demonstrates that multi-lingual approach does indeed aid model in learning each individual language better. Results 7 indicate a 4.0% increase in WRR for Hindi and a 0.20% increase for Gujarati.

7 Conclusion

We assembled a real dataset for STR in Indian languages. IndicSTR12 is the most comprehensive dataset available, and it is a first in many languages. We generated 3 million synthetic word-images for all 13 languages in addition to real datasets for faster STR solution development for Indic languages. In addition to benchmarking STR performance on three models, this paper establishes the need for even more data to leverage the learning powers of SOTA Latin models. Because Indian scripts are more complex than Latin scripts, they necessitate a more comprehensive training resource. In the future, this large dataset in both the real and synthetic domains will aid the Indic Scene text community in developing solutions for Indic STR that are on par with Latin solutions.

Acknowledgement. This work is supported by MeitY, Government of India, through the NLTM-Bhashini project.

References

1. Abnar, S., Zuidema, W.: Quantifying attention flow in transformers. arXiv preprint arXiv:2005.00928 (2020)
2. Baek, J., et al.: What is wrong with scene text recognition model comparisons? dataset and model analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4715–4723 (2019)
3. Baek, J., Matsui, Y., Aizawa, K.: What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3113–3122 (2021)
4. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
5. Bautista, D., Atienza, R.: Scene text recognition with permuted autoregressive sequence models. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13688, pp. 178–196. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19815-1_11
6. Bušta, M., Patel, Y., Matas, J.: E2E-MLT - an unconstrained end-to-end method for multi-language scene text. In: Carneiro, G., You, S. (eds.) ACCV 2018. LNCS, vol. 11367, pp. 127–143. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21074-8_11
7. Chandio, A.A., Asikuzzaman, M., Pickering, M., Leghari, M.: Cursive-text: a comprehensive dataset for end-to-end Urdu text recognition in natural scene images. Data Brief **31**, 105749 (2020)
8. Chen, X., Jin, L., Zhu, Y., Luo, C., Wang, T.: Text recognition in the wild: a survey. ACM Comput. Surv. (CSUR) **54**(2), 1–35 (2021)
9. Ch'ng, C.K., Chan, C.S., Liu, C.L.: Total-text: toward orientation robustness in scene text detection. Int. J. Doc. Anal. Recogn. (IJDAR) **23**(1), 31–52 (2020)
10. GOI: Government Indian language report (2011). https://censusindia.gov.in/census.website/
11. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376 (2006)
12. Gunna, S., Saluja, R., Jawahar, C.V.: Transfer learning for scene text recognition in Indian languages. In: Barney Smith, E.H., Pal, U. (eds.) ICDAR 2021. LNCS, vol. 12916, pp. 182–197. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86198-8_14
13. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2315–2324 (2016)
14. He, M., et al.: ICPR 2018 contest on robust reading for multi-type web images. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 7–12. IEEE (2018)
15. He, P., Huang, W., Qiao, Y., Loy, C.C., Tang, X.: Reading scene text in deep convolutional sequences. In: Thirtieth AAAI conference on artificial intelligence (2016)
16. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227 (2014)
17. Karatzas, D., et al.: ICDAR 2015 competition on robust reading. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1156–1160. IEEE (2015)
18. Karatzas, D., et al.: ICDAR 2013 robust reading competition. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1484–1493. IEEE (2013)
19. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Liu, W., Chen, C., Wong, K.Y.K., Su, Z., Han, J.: STAR-Net: a spatial attention residue network for scene text recognition. In: BMVC, vol. 2, p. 7 (2016)

21. Liu, Z., Li, Y., Ren, F., Goh, W.L., Yu, H.: SqueezedText: a real-time scene text recognition by binary convolutional encoder-decoder network. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
22. Lucas, S.M.: ICDAR 2003 robust reading competitions: entries, results, and future directions. IJDAR **7**, 105–122 (2005)
23. Mathew, M., Jain, M., Jawahar, C.: Benchmarking scene text recognition in Devanagari, Telugu and Malayalam. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 07, pp. 42–46 (2017). https://doi.org/10.1109/ICDAR.2017.364
24. Mathew, M., Singh, A.K., Jawahar, C.: Multilingual OCR for Indic scripts. In: 2016 12th IAPR Workshop on Document Analysis Systems (DAS), pp. 186–191. IEEE (2016)
25. Mishra, A., Alahari, K., Jawahar, C.: Scene text recognition using higher order language priors. In: BMVC-British Machine Vision Conference. BMVA (2012)
26. Nayef, N., et al.: ICDAR 2019 robust reading challenge on multi-lingual scene text detection and recognition-RRC-MLT-2019. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1582–1587. IEEE (2019)
27. Nayef, N., et al.: ICDAR 2017 robust reading challenge on multi-lingual scene text detection and script identification-RRC-MLT. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 1454–1459. IEEE (2017)
28. Phan, T.Q., Shivakumara, P., Tian, S., Tan, C.L.: Recognizing text with perspective distortion in natural scenes. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 569–576 (2013)
29. Risnumawan, A., Shivakumara, P., Chan, C.S., Tan, C.L.: A robust arbitrary text detection system for natural scene images. Expert Syst. Appl. **41**(18), 8027–8048 (2014)
30. Saluja, R., Maheshwari, A., Ramakrishnan, G., Chaudhuri, P., Carman, M.: OCR on-the-go: robust end-to-end systems for reading license plates & street signs. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 154–159. IEEE (2019)
31. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2298–2304 (2016)
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
33. Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. In: Artificial Intelligence and Machine Learning for Multi-domain Operations Applications, vol. 11006, pp. 369–386. SPIE (2019)
34. Su, B., Lu, S.: Accurate scene text recognition based on recurrent neural network. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9003, pp. 35–48. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16865-4_3
35. Sun, Y., Liu, J., Liu, W., Han, J., Ding, E., Liu, J.: Chinese street view text: large-scale Chinese text reading with partially supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9086–9095 (2019)
36. Sun, Y., et al.: ICDAR 2019 competition on large-scale street view text with partial labeling-RRC-LSVT. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1557–1562. IEEE (2019)
37. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
38. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-ResNet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)

39. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017). https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
40. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 International Conference on Computer Vision, pp. 1457–1464. IEEE (2011)
41. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
42. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: a 10 million image database for scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. **40**(6), 1452–1464 (2018). https://doi.org/10.1109/TPAMI.2017.2723009